



Red Hat Enterprise Linux 7

设备映射多路径

设备映射多路径（DM Multipath）配置及管理

Red Hat Enterprise Linux 7 设备映射多路径

设备映射多路径 (DM Multipath) 配置及管理

Steven Levine

Red Hat 客户文档服务

slevine@redhat.com

法律通告

Copyright © 2016 Red Hat, Inc. and others.

This document is licensed by Red Hat under the [Creative Commons Attribution-ShareAlike 3.0 Unported License](#). If you distribute this document, or a modified version of it, you must provide attribution to Red Hat, Inc. and provide a link to the original. If the document is modified, all Red Hat trademarks must be removed.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux ® is the registered trademark of Linus Torvalds in the United States and other countries.

Java ® is a registered trademark of Oracle and/or its affiliates.

XFS ® is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL ® is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js ® is an official trademark of Joyent. Red Hat Software Collections is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack ® Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

本指南提供在 Red Hat Enterprise Linux 7 中使用 DM Multipath 功能的相关信息。

目录

第 1 章 设备映射器多路径	3
1.1. 新的和更改的功能	3
1.2. DM MULTIPATH 概述	4
1.3. 存储阵列支持	6
1.4. DM MULTIPATH 组件	6
1.5. DM MULTIPATH 设置概述	7
第 2 章 多路径设备	8
2.1. 多路径设备识别符	8
2.2. 在集群中保持多路径设备名称一致	8
2.3. 多路径设备属性	9
2.4. 逻辑卷中的多路径设备	9
第 3 章 设置 DM MULTIPATH	11
3.1. 设置 DM MULTIPATH	11
3.2. 在生成多路径设备时忽略逻辑磁盘	12
3.3. 配置存储设备	14
3.4. 在 INITRAMFS 文件系统中设置启动 MULTIPATH	14
第 4 章 DM MULTIPATH 配置文件	16
4.1. 配置文件概述	16
4.2. 配置文件黑名单	17
4.3. 配置文件默认设置	19
4.4. 多路径设备配置属性	26
4.5. 配置文件设备	31
第 5 章 DM MULTIPATH 管理及故障排除	37
5.1. 使用多路径帮助程序 (MULTIPATH HELPER) 自动生成配置文件	37
5.2. 重新定义在线多路径设备大小	37
5.3. 将 ROOT 文件系统从单路径设备移动到多路径设备中	37
5.4. 将 SWAP 文件系统从单路径设备移动到多路径设备中	38
5.5. 多路径守护进程	39
5.6. 大量 LUN 造成的问题	39
5.7. 有 QUEUE_IF_NO_PATH 功能的问题	39
5.8. 多路径命令输出	40
5.9. 使用多路径命令进行多路径查询	40
5.10. 多路径命令选项	41
5.11. 使用 DMSETUP 命令确定设备映射器条目	41
5.12. MULTIPATHD 命令	42
5.13. 使用 MULTIPATHD 互动控制台进行故障排除	43
5.14. 删除软件包后清除多路径文件	43
附录 A. 修订历史	44
索引	45

第 1 章 设备映射器多路径

可使用 DM Multipath（Device Mapper Multipathing，设备映射多路径）将服务器节点和储存阵列间的多个 I/O 路径配置为一个单一设备。这些 I/O 路径是可包含独立电缆、交换器和控制器的实体 SAN 链接。多路径集合了 I/O 路径，并生成由这些集合路径组成的新设备。

本章提供了 Red Hat Enterprise Linux 7 初始发行版本之后增加的 DM-Multipath 新功能概述。除此之外，本章还提供 DM Multipath 及其组件的高级概述，以及 DM-Multipath 设置概述。

1.1. 新的和更改的功能

本小节列出了 Red Hat Enterprise Linux 7 最初发行本中的 DM Multipath 新功能。

1.1.1. Red Hat Enterprise Linux 7.1 版本的新功能和更改的功能

Red Hat Enterprise Linux 7.1 版本包括以下文档和功能的更新和更改。

- [表 5.1 “有用的 multipath 命令选项”](#)。现在增加了 `multipath` 命令的 `-w` 和 `-W` 选项，这使您可以更好地管理 `wwids` 文件。
- `multipath.conf` 文件中 `features` 参数的 `values` 有新增选项值，请参阅 [〈第 4 章 DM Multipath 配置文件〉](#)。
- [表 4.1 “多路径配置默认设置”](#) 包含 `force_sync` 参数条目，该条目可防止将 `async` 模式设定为 `"yes"` 后在该模式下运行路径检查程序。

除此之外，本文件中还有一些技术上的较小更正和阐述。

1.1.2. Red Hat Enterprise Linux 7.2 版本的新功能和更改的功能

Red Hat Enterprise Linux 7.2 版本包括以下文档和功能的更新和更改。

- 本文档包含一个新的小节 [第 5.1 节 “使用多路径帮助程序（Multipath Helper）自动生成配置文件”](#)。可通过 Multipath Helper 应用程序使用自定义别名、设备黑名单以及各个多路径设备的特别设置创建多路径配置。
- `multipath.conf` 配置文件的 `defaults` 部分支持新的 `config_dir`、`new_bindings_in_boot`、`ignore_new_boot_devs`、`retrigger_tries` 和 `retrigger_delays` 参数。有关 `multipath.conf` 文件的 `defaults` 部分的详情，请查看 [表 4.1 “多路径配置默认设置”](#)。
- `multipath.conf` 配置文件的 `defaults`、`devices` 和 `devices` 部分现在支持 `delay_watch_checks` 和 `delay_wait_checks` 参数。有关这些配置参数的详情，请参考 [第 4 章 DM Multipath 配置文件](#)。

除此之外，本文件中还有一些技术上的较小更正和阐述。

1.1.3. Red Hat Enterprise Linux 7.3 版本的新功能和更改的功能

Red Hat Enterprise Linux 7.3 版本包括以下文档和功能的更新和更改。

- `multipathd` 命令支持新格式命令，以便显示多路径状态以及使用 `"raw"` 格式版本的路径。在 `raw` 格式中，不会显示标头，且不会添加字段以便将标头与该列对其。反之，该字段以指定的格式显示。有关 `multipathd` 命令的详情，请查看 [第 5.12 节 “Multipathd 命令”](#)。

- 从 Red Hat Enterprise Linux 7.3 开始，如果在设备配置中指定 `prio "alua exclusive_pref_bit"`，多路径将生成包含只使用 `pref` 字节集的路径组，并为该组设定最高优先级。有关配置参数的详情，请查看 [第 4 章 DM Multipath 配置文件](#)。
- `multipath.conf` 配置文件的 `defaults`、`devices` 和 `devices` 部分现在支持 `skip_kpartx` 配置参数。有关这些配置参数的详情，请参考 [第 4 章 DM Multipath 配置文件](#)。

除此之外，本文件中还有一些技术上的较小更正和阐述。

1.2. DM MULTIPATH 概述

可使用 DM Multipath 提供以下功能：

- 冗余

DM Multipath 能够在主动/被动配置下提供故障转移。在主动/被动配置下，只有一半的路径在每次进行 I/O 时会被使用。若一条 I/O 路径的任一元素（电缆、交换器或者控制器）出现故障，DM Multipath 就会将它切换到备用路径。

- 改进的性能

可将 DM Multipath 配置为主动/主动模式，这会将 I/O 以轮循机制（round-robin）的方式分布到所有的路径中。在有些配置中，DM Multipath 能够检测 I/O 路径的负载，并且重新动态平衡负载。

图 1.1 “带一个 RAID 设备的主动/被动多路径配置”演示了在服务器和 RAID 设备之间有两个 I/O 路径的主动/被动配置。这里服务器中有两个 HBA，两个 SAN 交换机以及两个 RAID 控制器。

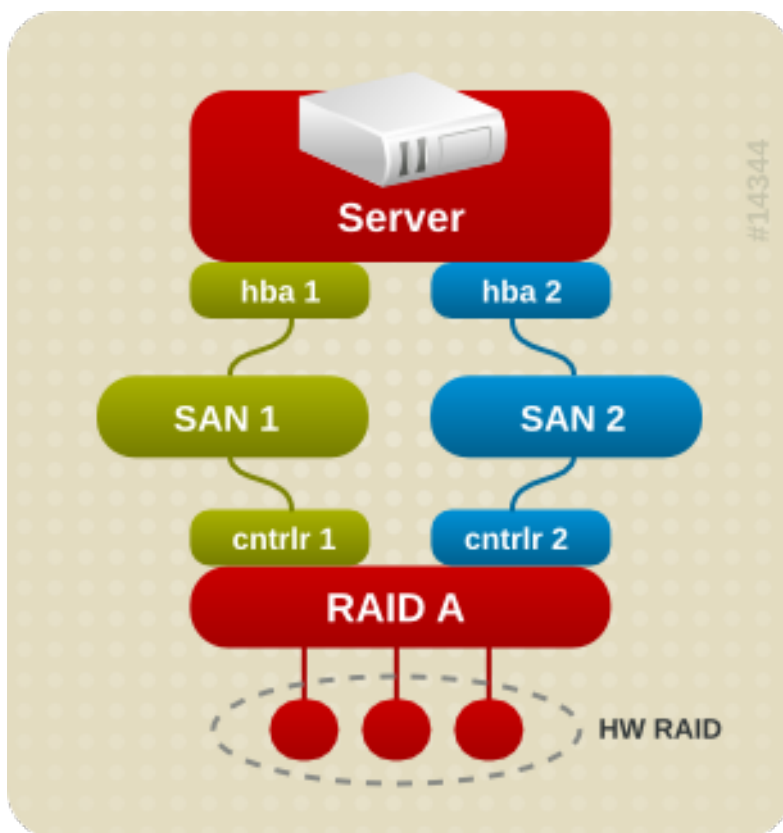


图 1.1. 带一个 RAID 设备的主动/被动多路径配置

在这个配置中，一个 I/O 路径通过 hba1、SAN1 以及控制器 1。另一个 I/O 路径通过 hba2、SAN2 以及控制器 2。在这个配置中有很多地方可能出现故障：

- HBA 故障
- FC 电缆故障
- SAN 交换机故障
- 阵列控制器端口故障

阵列控制器端口故障

图 1.2 “带两个 RAID 设备的主动/被动多路径配置” 演示了更复杂的主动/被动配置，其中服务器中有两个 HBA、两个 SAN 交换机以及两个 RAID 设备（每个带两个 RAID 控制器）。

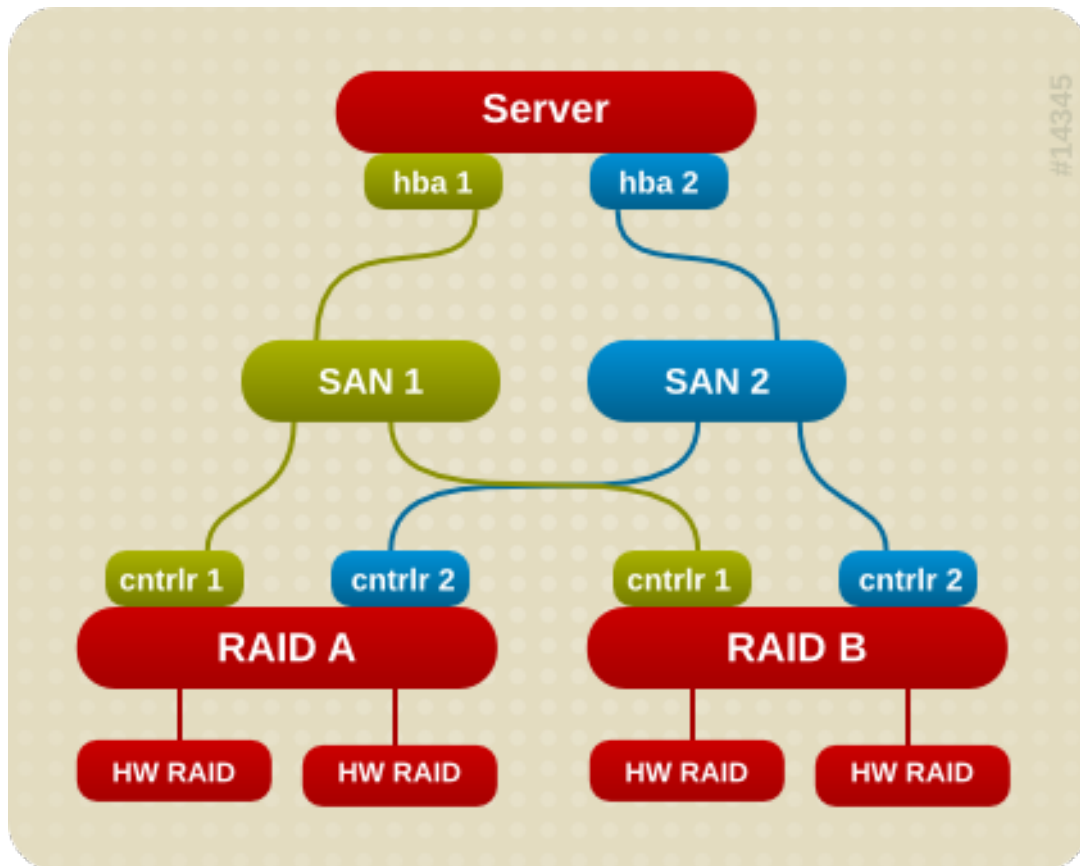


图 1.2. 带两个 RAID 设备的主动/被动多路径配置

在图 1.2 “带两个 RAID 设备的主动/被动多路径配置” 演示的示例中，每个 RAID 设备都有两个 I/O 路径（与图 1.1 “带一个 RAID 设备的主动/被动多路径配置” 的示例相同）。如果配置了 DM Multipath，任一 RAID 设备的 I/O 路径的任一点出现故障，都将会导致 DM Multipath 为该设备切换到备用 I/O 路径。

图 1.3 “带一个 RAID 设备的主动/主动多路径配置” 演示在服务器中有两个 HBA、一个 SAN 交换机以及两个 RAID 控制器的主动/主动配置。在服务器和存储设备间有四条 I/O 路径：

- hba1 到控制器 1
- hba1 到控制器 2
- hba2 到控制器 1
- hba2 到控制器 2

在这个配置中，可将 I/O 分布到上述四条路径中。



图 1.3. 带一个 RAID 设备的主动/主动多路径配置

1.3. 存储阵列支持

默认情况下，DM Multipath 支持大多数常用的支持 DM Multipath 的储存阵列。若要了解默认配置值和支持的设备的相关信息，请运行下列指令中的任何一个。

```
# multipathd show config
# multipath -t
```

若储存阵列支持 DM Multipath 但是默认情况下未被配置，您需要将它加入 DM Multipath 配置文件 `multipath.conf`。如需了解 DM Multipath 配置文件的相关信息，请参阅 [〈第 4 章 DM Multipath 配置文件〉](#)。

有些存储阵列对 I/O 错误和路径切换需要特殊处理。这些都需要独立的硬件处理器 kernel 模块。

1.4. DM MULTIPATH 组件

表 1.1 “DM Multipath 组件”对 DM Multipath 组件进行了描述。

表 1.1. DM Multipath 组件

组件	描述
<code>dm-multipath</code> kernel 模块	为路径和路径组重新指定 I/O 并支持故障转移。
<code>mpathconf</code> 工具程序	配置并启用 DM Multipath。

组件	描述
multipath 指令	列出并配置多路径设备。通常使用 <code>/etc/rc.sysinit</code> 启动，还可以在添加块设备时通过 udev 程序启动。
multipathd 守护进程	监视路径；若路径故障并返回，它可能会启动路径组切换。允许对多路径设备进行交互式修改。若需对 <code>/etc/multipath.conf</code> 文件进行任何修改，都必须重新启动本守护进程。
kpartx 命令	为设备分区生成设备映射器。对于带 DM Multipath 的基于 DOS 的分区来说，使用此命令很有必要。 kpartx 命令包含在它自己的软件包当中，但是 device-mapper-multipath 软件包需要依赖它。

1.5. DM MULTIPATH 设置概述

DM Multipath 包含适用于常用多路径配置的已编译默认设置。为系统配置 DM Multipath 的基本步骤如下：

1. 安装 **device-mapper-multipath** rpm。
2. 使用 **mpathconf** 命令创建配置文件并启用多路径。若无需编辑该配置文件，可使用此命令启动多路径守护程序。
3. 如需编辑该配置文件，请编辑 **multipath.conf** 配置文件，修改默认值并保存更新的文件。
4. 启动多路径守护进程。

如需了解多路径配置的安装说明，请参阅 [〈第 3 章 设置 DM Multipath〉](#)。

第 2 章 多路径设备

若没有 DM Multipath，从服务器节点到储存控制器的每一条路径都会被系统视为独立的设备，即使 I/O 路径连接的是相同的服务器节点到相同的储存控制器也是如此。DM Multipath 提供了有逻辑的管理 I/O 路径的方法，即在基础设备顶端生成单一多路径设备。

2.1. 多路径设备识别符

每个多路径设备都有一个全球识别符（WWID），它是一个全球唯一的无法更改的号码。默认情况下会将多路径设备的名称设定为它的 WWID。另外，您还可以在多路径配置文件中设置 `user_friendly_names` 选项，该选项可将别名设为格式为 `mpathn` 的节点唯一名称。

例如：有两个 HBA 的节点通过单一不分区 FC 交换机附加到有两个端口的储存控制器中时，可看到四个设备：`/dev/sda`、`/dev/sdb`、`dev/sdc` 以及 `/dev/sdd`。DM Multipath 会生成由唯一 WWID 的单一设备，该设备可根据多路径配置将 I/O 重新路由到那四个基础设备。`user_friendly_names` 配置选项的值被设为 `yes` 时，多路径设备的名称会被设定为 `mpathn`。

新设备被纳入 DM Multipath 控制时，该设备会显示在 `/dev` 目录的两个不同位置：`/dev/mapper/mpathn` 和 `/dev/dm-n`。

- `/dev/mapper` 中的设备是在引导过程中生成的。可使用这些设备访问多路径设备，例如在生成逻辑卷时。
- 任何 `/dev/dm-n` 格式的设备都只适用于内部使用，管理员请勿直接使用。

有关多路径配置默认的详情，包括 `user_friendly_names` 配置选项，请参考第 4.3 节“配置文件默认设置”。

您还可以使用多路径配置文件 `multipaths` 部分的 `alias` 选项将多路径设备名称设为您选择的名称。有关多路径配置文件 `multipaths` 部分的详情请参考第 4.4 节“多路径设备配置属性”。

2.2. 在集群中保持多路径设备名称一致

当将 `user_friendly_names` 配置选项设为 `yes` 时，该多路径设备的名称对于节点来说是唯一的，但不保证对使用多路径设备的所有节点都一致。同样，如果您为 `multipath.conf` 配置文件的 `multipaths` 部分中的设备设定 `alias` 选项，该名称不会自动在集群的所有节点中保持一致。如果您使用 LVM 在多路径设备中创建逻辑设备，这不应是问题。但如果您需要将您的多路径设备名称在集群中的每个节点上都保持一致，请不要将 `user_friendly_names` 选项设定为 `yes`，且不要为那些设备配置别名。默认情况下，如果您不将 `user_friendly_names` 设定为 `yes`，或者为某个设备配置别名，则设备名称将是该设备的 WWID，它是不会变的。

如果您要系统定义的用户友好名称在集群的所有节点中都一致，您可按照以下步骤操作：

1. 在一台机器中设定所有多路径设备。
2. 运行以下命令禁用其它机器上所有的多路径设备：

```
# service multipathd stop
# multipath -F
```

3. 将第一台机器中的 `/etc/multipath/bindings` 文件复制到集群中的其它所有机器中。
4. 使用以下命令在集群的其他机器中重新 `multipathd` 守护进程：

-

```
# service multipathd start
```

如果您添加新设备，您将需要重复这个过程。

同样，如果您为某个设备配置别名以便在集群的节点中使其保持一致，您应确定 `/etc/multipath.conf` 文件对于集群中的每个节点都是一样的，步骤如下：

1. 为一台机器上 `multipath.conf` 文件中的多路径设备配置别名。
2. 运行以下命令禁用其它机器上所有的多路径设备：

```
# service multipathd stop
# multipath -F
```

3. 将第一台机器中的 `/etc/multipath.conf` 文件复制到集群中的其它所有机器中。
4. 使用以下命令在集群的其他机器中重新 `multipathd` 守护进程：

```
# service multipathd start
```

当您添加新设备时，您将需要重复这个过程。

2.3. 多路径设备属性

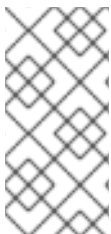
除 `user_friendly_names` 和 `alias` 选项外，`multipath` 设备有大量属性。您可以为具体 `multipath` 设备修改这些属性，方法是在 `multipath` 配置文件的 `multipaths` 部分为那个设备创建一个条目。有关 `multipath` 配置文件 `multipaths` 部分的详情请参考第 4.4 节“多路径设备配置属性”。

2.4. 逻辑卷中的多路径设备

生成多路径设备后，多路径设备的名称可以与您生成一个 LVM 物理卷时使用的物理设备名称相同。例如：如果多路径名称为 `/dev/mapper/mpatha`，以下命令可将 `/dev/mapper/mpatha` 标记为一个物理卷。

```
pvcreate /dev/mapper/mpatha
```

您可以如在使用其它 LVM 物理设备一样在创建 LVM 卷组时使用得到的 LVM 物理卷。



注意

如果您要在配置了分区的整个设备中创建 LVM 物理卷，`pvcreate` 命令将会失败。请注意：如果您不具体指定每个块设备，Anaconda 和 Kickstart 安装程序会生成空分区表。您可以使用 `kpartx -d` 和 `fdisk` 命令删除现有分区。如果您的系统有大于 2 TB 的块设备，您可使用 `parted` 命令删除分区。

当您创建使用主动/被动多路径阵列的 LVM 逻辑卷作为基础物理设备时，您应该在 `/etc/lvm/lvm.conf` 文件中添加过滤器，以排除构成多路径设备的磁盘。这是因为如果不过滤，阵列在接收到 I/O 时自动从主动路径切换到被动路径，只要 LVM 扫描到被动路径，多路径将会进行故障转移和自动恢复。对于需要使用命令激活被动路径的主动/被动阵列，LVM 会给出警告信息。

要过滤 LVM 配置文件 (`lvm.conf`) 中的所有 SCSI 设备, 请在该文件的 **devices** 部分添加以下过滤器。

```
filter = [ "r/block/", "r/disk/", "r/sd.*/", "a/.*/" ]
```

第 3 章 设置 DM MULTIPATH

本章提供了配置 DM Multipath 的分步示例，其中包括以下步骤：

- 基本 DM Multipath 设置
- 忽略本地磁盘
- 在配置文件中添加更多设备
- 在 `initramfs` 文件系统中启动 Multipath

3.1. 设置 DM MULTIPATH

安装 DM-Multipath 前，请确定系统已升级，其中包括 `device-mapper-multipath` 软件包升级。

可使用 `mpathconf` 程序设置多路径，它可创建多路径配置文件 `/etc/multipath.conf`。

- 如果 `/etc/multipath.conf` 文件已存在，`mpathconf` 程序将会编辑该文件。
- 如果 `/etc/multipath.conf` 文件不存在，`mpathconf` 程序将使用 `/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf` 文件作为起始文件。
- 如果 `/usr/share/doc/device-mapper-multipath-0.4.9/multipath.conf` 文件不存在，则 `mpathconf` 程序将从头开始创建 `/etc/multipath.conf` 文件。

如果不需要编辑 `/etc/multipath.conf` 文件，则可以运行以下命令为 DM-Multipath 设定基本故障切换配置。这个命令可启用多路径配置文件并启动 `multipathd` 守护进程。

```
# mpathconf --enable --with_multipathd y
```

如果需要在启动 `multipathd` 守护进程前编辑 `/etc/multipath.conf` 文件，请执行以下步骤为 DM-Multipath 设置基本故障切换配置。

1. 运行带 `--enable` 选项的 `mpathconf` 命令：

```
# mpathconf --enable
```

有关可能需要的 `mpathconf` 命令的其它选项，请参考 `mpathconf` 手册页，或者运行指定 `--help` 选项的 `mpathconf` 命令。

```
# mpathconf --help
usage: /sbin/mpathconf <command>

Commands:
Enable: --enable
Disable: --disable
Set user_friendly_names (Default y): --user_friendly_names <y|n>
Set find_multipaths (Default y): --find_multipaths <y|n>
Load the dm-multipath modules on enable (Default y): --with_module
<y|n>
start/stop/reload multipathd (Default n): --with_multipathd <y|n>
```

2. 必要时请编辑 `/etc/multipath.conf` 文件。会将 DM Multipath 的默认设置编译到系统中，且无需在 `/etc/multipath.conf` 文件中明确设定。

`path_grouping_policy` 的默认值是设为 **failover**，因此在这个示例中您不需要编辑 `/etc/multipath.conf` 文件。有关在配置文件中将默认值改为其它数值的详情请参考 [第 4 章 DM Multipath 配置文件](#)。

在您系统配置文件的初始默认部分配置您的系统以便多路径设备的名称格式为 `mpathn`。如果没有这个设置，多路径设备的名称将会是该设备 WWID 别名。

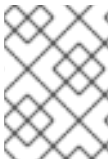
3. 需要时保存配置文件并退出编辑器。
4. 执行以下命令：

```
# service multipathd start
```

因为在配置文件中将 `user_friendly_name` 值设为 **yes**，那么生成的多路径设备将为 `/dev/mapper/mpathn`。有关将设备名称设定为您选择的别名的方法，请参考 [第 4 章 DM Multipath 配置文件](#)。

如果不想使用用户友好名称，可以运行以下命令：

```
# mpathconf --enable --user_friendly_names n
```



注意

如果在启动 multipath 守护进程后发现需要编辑 multipath 配置文件，则必须执行 **service multipathd reload** 命令方可使更改生效。

3.2. 在生成多路径设备时忽略逻辑磁盘

有些机器在其内部磁盘中有本地 SCSI 卡。不建议在这些设备中使用 DM Multipath。如果将 `find_multipaths` 配置参数设定为 **yes**，则不一定要将这些设备列入黑名单。有关 `find_multipaths` 配置参数的详情，请参考 [第 4.3 节“配置文件默认设置”](#)。

如果没有将 `find_multipaths` 配置参数设定为 **yes**，则可以使用以下步骤修改多路径配置文件，以便在配置多路径时忽略本地磁盘。

1. 确定哪些磁盘是内部磁盘并将其列入黑名单。

在这个示例中，`/dev/sda` 是内部磁盘。注：因为在默认多路径配置文件中已经对其进行了配置，所以执行 `multipath -v2` 会在多路径映射中显示本地磁盘 `/dev/sda`。

有关 `multipath` 命令输出的详情请参考 [第 5.8 节“多路径命令输出”](#)。

```
# multipath -v2
create: SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1 undef
WINSYS,SF2372
size=33 GB features="0" hwhandler="0" wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 0:0:0:0 sda 8:0 [-----]

device-mapper ioctl cmd 9 failed: Invalid argument
```



```

device-mapper ioctl cmd 14 failed: No such device or address
create: 3600a0b80001327d80000006d43621677 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:0 sdb 8:16 undef ready running
  `-- 3:0:0:0 sdf 8:80 undef ready running

create: 3600a0b80001327510000009a436215ec undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:1 sdc 8:32 undef ready running
  `-- 3:0:0:1 sdg 8:96 undef ready running

create: 3600a0b80001327d800000070436216b3 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:2 sdd 8:48 undef ready running
  `-- 3:0:0:2 sdg 8:112 undef ready running

create: 3600a0b80001327510000009b4362163e undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:3 sdd 8:64 undef ready running
  `-- 3:0:0:3 sdg 8:128 undef ready running

```

2. 为防止设备映射器将 `/dev/sda` 与其多路径对应，请编辑 `/etc/multipath.conf` 文件的 `blacklist` 部分，使其包括该设备。虽然可以使用 `devnode` 类型将 `sda` 设备列入黑名单，但并不安全，因为重启时无法保证 `/dev/sda` 使用同一个名称。要将单独的设备列入黑名单，可以使用那个设备的 WWID 将其列入黑名单。

请注意：在 `multipath -v2` 命令的输出中，`/dev/sda` 设备的 WWID 是 `SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1`。要将这个设备列入黑名单，请在 `/etc/multipath.conf` 文件中添加以下内容。

```

blacklist {
    wwid SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1
}

```

3. 更新 `/etc/multipath.conf` 文件后，必须手动让 `multipathd` 守护进程重新载入该文件。以下命令可重新载入更新的 `/etc/multipath.conf` 文件。

```
# service multipathd reload
```

4. 运行以下命令删除多路径设备：

```
# multipath -f SIBM-ESXSST336732LC____F3ET0EP0Q000072428BX1
```

5. 要查看是否删除了该设备，可以输入 `multipath -ll` 命令显示当前的多路径配置。有关 `multipath -ll` 命令的详情，请参考第 5.9 节“使用多路径命令进行多路径查询”。

要确定被列入黑名单的设备没有被重新添加回来，可输入 `multipath` 命令，如下所示。如果没有指定 `-v` 选项，则 `multipath` 命令默认详细等级为 `v2`。

```
# multipath

create: 3600a0b80001327d80000006d43621677 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:0 sdb 8:16 undef ready running
  ` - 3:0:0:0 sdf 8:80 undef ready running

create: 3600a0b80001327510000009a436215ec undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:1 sdc 8:32 undef ready running
  ` - 3:0:0:1 sdg 8:96 undef ready running

create: 3600a0b80001327d800000070436216b3 undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:2 sdd 8:48 undef ready running
  ` - 3:0:0:2 sdg 8:112 undef ready running

create: 3600a0b80001327510000009b4362163e undef WINSYS,SF2372
size=12G features='0' hwhandler='0' wp=undef
`-+- policy='round-robin 0' prio=1 status=undef
  |- 2:0:0:3 sdd 8:64 undef ready running
  ` - 3:0:0:3 sdg 8:128 undef ready running
```

3.3. 配置存储设备

默认情况下，DM Multipath 支持大多数常用的支持 DM Multipath 的储存阵列。若要了解默认配置值和支持的设备的相关信息，请运行下列指令中的任意一个。

```
# multipathd show config
# multipath -t
```

如果需要添加一个在默认情况下不支持的存储设备作为已知多路径设备，请编辑 `/etc/multipath.conf` 文件，并插入正确的设备信息。

例如：要添加有关 HP Open-V 系列的信息（其条目示例如下）。在这个示例中将设备设定在所有路径均失败后排队 1 分钟（或 12 次尝试，每 5 秒后重试一次）。

```
devices {
    device {
        vendor "HP"
        product "OPEN-V"
        no_path_retry 12
    }
}
```

有关配置文件 `devices` 部分的详情请参考 [第 4.5 节“配置文件设备”](#)。

3.4. 在 INITRAMFS 文件系统中设置启动 MULTIPATH

可以在 **initramfs** 文件系统中设定启动 multipath。配置 multipath 后，可使用该 multipath 配置文件重建 **initramfs** 文件系统，方法是执行附带以下选项的 **dracut** 命令。

```
# dracut --force --add multipath --include /etc/multipath
```

如果在 **initramfs** 文件系统中运行 multipath，并更改 multipath 配置文件，则必须重建 **initramfs** 文件系统方可使更改生效。

第 4 章 DM MULTIPATH 配置文件

DM Multipath 默认为大多数常用多路径提供配置值。此外，DM Multipath 还包括对那些本身支持 DM Multipath 的常见储存阵列的支持。如需了解有关默认配置值以及支持的设备的相关信息，请运行下列命令之一。

```
# multipathd show config
# multipath -t
```

可以编辑 `/etc/multipath.conf` 配置文件覆盖 DM Multipath 的默认配置值。必要时可在配置文件中添加默认不支持的储存阵列。



注意

可以在 `initramfs` 文件系统中运行 `set up multipathing`。如果在 `initramfs` 文件系统中运行 `multipath`，并可以更改 `multipath` 配置文件，则必须重建 `initramfs` 文件系统以使更改生效。有关重建使用 `multipath` 的 `initramfs` 文件系统的详情，请参考 [第 3.4 节“在 initramfs 文件系统中设置启动 Multipath”](#)。

本章提供了解析和修改 `multipath.conf` 文件的详情。它包含以下小节：

- 配置文件概述
- 配置文件黑名单
- 配置文件默认
- 配置文件多路径
- 配置文件设备

在 `multipath` 配置文件中，只需指定您的配置需要的部分，或者是您想要修改的默认值即可。如果文件中有和您的环境不相关的部分，或者是不需要覆盖的默认值，可以将它们注释出来，因为它们位于初始文件中。

配置文件使用正则表达式描述语法。

关于配置文件的详细信息，请参阅 `multipath.conf(5)` 手册页。

4.1. 配置文件概述

多路径配置文件可分为以下几个部分：

blacklist

不被视为多路径的具体设备列表。

blacklist_exceptions

根据 `blacklist` 部分中的参数列出不在黑名单中的多路径设备。

defaults

DM Multipath 的常规默认设置。

multipaths

各个独立多路径设备的特性设置。这些数值覆盖了在配置文件的 **defaults** 和 **devices** 部分中指定的数值。

devices

各个存储控制器的设置。这些数值覆盖了在配置文件的 **defaults** 部分指定的数值。如果要使用不是默认支持的存储阵列，则可能需要为您的阵列创建 **devices** 子部分。

系统决定多路径设备的属性时，会先检查多路径设置，然后检查设备设置，最后才检查多路径系统默认设置。

4.2. 配置文件黑名单

多路径配置文件的 **blacklist** 部分指定在系统配置多路径设备时不能使用的设备。黑名单中的设备将无法分组到多路径设备中。

在旧版 Red Hat Enterprise Linux 中，multipath 总是尝试为每个没有明确列入黑名单的路径创建多路径设备。但在 Red Hat Enterprise Linux 6 中，如果 **find_multipaths** 配置参数被设定为 **yes**，multipath 将只在满足以下三个条件之一时创建设备：

- 至少有两个使用同一 WWID 的路径没有被列入黑名单。
- 用户可使用 **multipath** 命令手动强制创建该设备。
- 有与之前创建的多路径设备相同 WWID 的路径（即使那个多路径设备目前不存在）。无论何时，创建多路径设备后，多路径会记住该设备的 WWID，以便在它看到有使用那个 WWID 的路径时即自动再次创建该设备。这可允许您让多路径自动选择正确的路径以便创建多路径设备而无需编辑多路径黑名单。

如果您之前创建了一个 multipath 设备而没有使用 **find_multipaths** 参数，然后您随后将该参数设定为 **yes**，您需要删除所有设备的 WWID，您不想从 **/etc/multipath/wwids** 文件中将其创建为 multipath 设备。以下显示示例 **/etc/multipath/wwids** 文件。WWID 由斜线 (/) 括起来：

```
# Multipath wwids, Version : 1.0
# NOTE: This file is automatically maintained by multipath and
multipathd.
# You should not need to edit this file in normal circumstances.
#
# Valid WWIDs:
/3600d0230000000000e13955cc3757802/
/3600d0230000000000e13955cc3757801/
/3600d0230000000000e13955cc3757800/
/3600d02300069c9ce09d41c31f29d4c00/
/SWINSYS SF2372 0E13955CC3757802/
/3600d0230000000000e13955cc3757803/
```

如将 **find_multipaths** 参数设定为 **yes**，则您只需要将那些您不想使其具有多路径功能的多路径设备列入黑名单，通常不需要将设备列入黑名单。

如果您需要将设备列入黑名单，您可以根据以下条件进行选择：

- 根据 WWID，如 [第 4.2.1 节“根据 WWID 将设备列入黑名单”](#) 所述

- 根据设备名称，如 [第 4.2.2 节 “根据设备名称将设备列入黑名单”](#) 所述
- 根据设备类型，如 [第 4.2.3 节 “根据设备类型将其加入黑名单”](#) 所述

默认情况下，各种设备类型都是列在黑名单中的，即使您将配置文件的初始黑名单部分注释出来也是如此。详情请参考 [第 4.2.2 节 “根据设备名称将设备列入黑名单”](#)。

4.2.1. 根据 WWID 将设备列入黑名单

您可以根据全球识别符将单独的设备列入黑名单，即在配置文件的 **blacklist** 部分加入 **wwid** 条目。

以下示例显示了在配置文件中可将 WWID 为 26353900f02796769 的设备列入黑名单的行。

```
blacklist {
    wwid 26353900f02796769
}
```

4.2.2. 根据设备名称将设备列入黑名单

您可以根据设备名称将设备类型列入黑名单，以便在配置文件 **blacklist** 部分的 **devnode** 条目中指定不要将它们分组到多路径设备中。

以下实例显示该配置文件中的可以将所有 SCSI 设备放入黑名单的行，因为它将所有 sd* 设备放入黑名单。

```
blacklist {
    devnode "^sd[a-z]"
}
```

您可以使用配置文件 **blacklist** 部分的 **devnode** 条目指定要列入黑名单的每个设备，而不是指定具体类型的所有设备，我们不推荐使用后一种情况。除非根据 **udev** 规则的静态映射，我们无法保证在重启后每个设备的名称是一样的。例如：重启后，某个设备的名称可能从 **/dev/sda** 变为 **/dev/sdb**。

默认情况下，以下 **devnode** 条目将会被列入默认黑名单中；因为这些条目而被列入黑名单的设备通常不支持 DM Multipath。如需在这些设备中启用 multipath，则需要在配置文件的 **blacklist_exceptions** 部分指定这些设备，详情请参阅 [第 4.2.4 节 “黑名单之外的设备”](#)。

```
blacklist {
    devnode "^(ram|raw|loop|fd|md|dm-|sr|scd|st)[0-9]*"
    devnode "^(td|ha)d[a-z]"
}
```

4.2.3. 根据设备类型将其加入黑名单

您可以在配置文件的 **blacklist** 部分与 **device** 一同指定具体设备类型。以下实例将所有 IBM DS4200 和 HP 设备放入黑名单。

```
blacklist {
    device {
        vendor "IBM"
        product "3S42"          #DS4200 Product 10
    }
    device {
```

```

        vendor "HP"
        product "*"
    }
}

```

4.2.4. 黑名单之外的设备

您可使用配置文件的 **blacklist_exceptions** 部分为被默认加入黑名单的设备启用多路径。

例如：如果您有大量设备，但只有一个需要多路径（WWID 为 3600d0230000000000e13955cc3757803），您不需要将您想要使用多路径的设备之外的每个设备单独加入黑名单，您只需要将所有设备都加入黑名单，然后在 `/etc/multipath.conf` 文件中添加以下行以便只允许您想要使用多路径的设备：

```

blacklist {
    wwid "*"
}

blacklist_exceptions {
    wwid "3600d0230000000000e13955cc3757803"
}

```

当在配置文件的 **blacklist_exceptions** 指定设备时，您必须以指定黑名单的相同方法来指定例外情况。例如：在 **devnode** 黑名单条目中指定的设备无法使用 WWID 将其指定为例外情况，即使列入黑名单的设备和该 WWID 关联也不行。同样，**devnode** 例外也只适用于 **devnode** 条目，而 **device** 例外只适用于 **device** 条目。

4.3. 配置文件默认设置

`/etc/multipath.conf` 配置文件包括 **defaults** 部分，在该部分中将 **user_friendly_names** 参数设为 **yes**，如下。

```

defaults {
    user_friendly_names yes
}

```

这可覆盖 **user_friendly_names** 参数的默认值。

该配置文件包括配置默认模板。这部分要被注释出来，如下。

```

#defaults {
#    polling_interval          10
#    path_selector              "round-robin 0"
#    path_grouping_policy      multibus
#    uid_attribute              ID_SERIAL
#    prio                       alua
#    path_checker               readsector0
#    rr_min_io                  100
#    max_fds                    8192
#    rr_weight                   priorities
#    failback                    immediate
#    no_path_retry               fail
#    user_friendly_names        yes
#}

```

要覆盖任意配置参数的默认值，您可将这个模板中相关的行复制到 **defaults** 部分并取消其注释。例如：要覆盖 **path_grouping_policy** 参数以使用 **multibus** 覆盖默认的 **failover**，请将模板中正确的行复制到配置文件的 **defaults** 部分并取消对它的注释，如下。

```
defaults {
    user_friendly_names    yes
    path_grouping_policy   multibus
}
```

表 4.1 “多路径配置默认设置” 描述了 **multipath.conf** 配置文件的 **defaults** 部分中设置的属性。DM Multipath 会使用这些值，除非该属性被 **multipath.conf** 文件的 **devices** 和 **multipaths** 部分所指定的属性覆盖。

表 4.1. 多路径配置默认设置

属性	描述
polling_interval	以秒为单位指定两次路径检查之间的间隔。对正常工作的路径，两次检查间的间隔会逐渐增加到 polling_interval 的四倍。默认值为 5。
multipath_dir	保存动态共享对象的目录。默认值依系统而定，通常为 /lib/multipath 。
find_multipaths	<p>定义设定多路径设备的模式。如果将这个参数设定为 yes，多路径将不会尝试为不在黑名单中的每个路径都创建设备，而是只在满足以下三个条件之一时才创建设备：</p> <ul style="list-style-type: none"> - 至少有两个路径没有使用同一 WWID 列入黑名单。 - 用户通过使用 multipath 命令指定设备强制创建该设备。 - 路径拥有与之前创建的多路径设备相同的 WWID。无论何时使用 find_multipaths 组件创建多路径设备后，多路径都会记住该设备的 WWID 以便在再次看到使用那个 WWID 的路径时自动创建设备。这可让您自动选择正确的所路径创建多路径设备而无需编辑多路径黑名单。如果之前您在没有设定 find_multipaths 参数的情况下创建了多路径设备，有关步骤请查看 第 4.2 节 “配置文件黑名单”。 <p>默认值为 no。默认 multipath.conf 文件由 mpathconf 命令生成，但从 Red Hat Enterprise Linux 7 开始才启用 find_multipaths。</p>
reassign_maps	启用 device-mapper 映射的创新分配。使用这个选项后， multipathd 守护进程将重新映射现有 device-mapper 映射，使其永远指向多路径设备，而不是基础块设备。可能的值包括 yes 和 no 。默认值为 yes 。
verbosity	默认详情。数值越高则详细程度越高。有效等级在 0 - 6 之间。默认值为 2。

属性	描述
path_selector	<p>指定用来决定下一个 I/O 操作所使用路径的默认算法。可能的值包括：</p> <p>round-robin 0：在路径组中循环每个路径，向每个路径发送同样数量的 I/O。</p> <p>queue-length 0：将下一组 I/O 发送到具有最少未处理 I/O 请求的路径。</p> <p>service-time 0：将下一组 I/O 发送到具有最短预计服务时间的路径，这是由未处理 I/O 的总量除以每个路径的相对流量决定的。</p> <p>默认值为 round-robin 0。</p>
path_grouping_policy	<p>指定用于未指定路径的默认路径分组策略，可能的值包括：</p> <p>failover：每个优先组群有一个路径。</p> <p>multibus：所有有效路径在一个优先组群中。</p> <p>group_by_serial：每个检测到的系列号有一个优先组群。</p> <p>group_by_prio：每个优先组群有一个路径优先值。优先权根据指定为 <code>global</code>、<code>per-controller</code> 或者 <code>per-multipath</code> 选项的调用程序决定。</p> <p>group_by_node_name：每个目标节点名有一个优先组。目标节点名保存在 <code>/sys/class/fc_transport/target*/node_name</code> 文件中。</p> <p>默认值为 failover。</p>

属性	描述
prio	<p>指定要获得路径优先值所需调用的默认程序及参数。例如：SPC-3 中的 ALUA 字节提供了一个可改变的 prio 值。可能的值包括：</p> <p>const：为所有路径设定优先权 1。</p> <p>emc：为 EMC 阵列生成路径优先权。</p> <p>alua：根据 SCSI-3 ALUA 设置生成路径有限级。从 Red Hat Enterprise Linux 7.3 开始，如果在设备配置中指定 prio "alua exclusive_pref_bit"，多路径将生成包含只使用 pref 字节集的路径组，并为该组设定最高优先级。</p> <p>ontap：为 NetApp 阵列生成路径优先权。</p> <p>rdac：为 LSI/Engenio RDAC 控制程序生成路径优先权。</p> <p>hp_sw：为 Compaq/HP 控制程序在激活/等待模式中生成路径优先权。</p> <p>hds：为 Hitachi HDS Modular 存储阵列生成路径优先权。</p> <p>默认值为 const。</p>
features	<p>多路径设备的默认额外功能，使用以下格式：<i>"number_of_features_plus_arguments feature1 ..."</i>。</p> <p>features 可能的值包括：</p> <p>queue_if_no_path，与将 no_path_retry 设置为 queue 相同。欲了解使用此功能可能出现的问题，请参阅 〈第 5.7 节“有 queue_if_no_path 功能的问题”〉。</p> <p>retain_attached_hw_handler：若此参数值被设为 yes，并且 SCSI 层已经为路径设备添加了硬件控制程序，multipath 将不会强制设备使用 multipath.conf 文件指定的 hardware_handler。若 SCSI 层没有添加硬件控制程序，multipath 仍然会使用已配置的硬件控制程序。默认值为 no。</p> <p>pg_init_retries n：路径组初始化重试，失败前最多重试 <i>n</i> 次，$1 \leq n \leq 50$。</p> <p>pg_init_delay_msecs n：在路径组初始化重试的间隔，等待 <i>n</i> 毫秒，$0 \leq n \leq 60000$。</p>

属性	描述
path_checker	<p>指定用于决定路径状态的默认方法，可能的值包括：</p> <p>readsector0：读取该设备的第一扇区。</p> <p>tur：在设备中执行 TEST UNIT READY 命令。</p> <p>emc_clariion：查询 EMC Clariion 具体 EVPD 页面 0xCO 以便决定路径。</p> <p>hp_sw：为使用 Active/Standby 固件的 HP 存储阵列检查路径状态。</p> <p>rdac：检查 LSI/Engenio RDAC 储存控制器的路径状态。</p> <p>directio：使用直接 I/O 读取第一个扇区。</p> <p>默认值为 directio。</p>
failback	<p>管理路径组群出错切换。</p> <p>immediate 值指定立即恢复到包含活跃路径的最高级别路径组群。</p> <p>manual 值指定不需要立即恢复，只有在操作者干预的情况下会发生恢复。</p> <p>followover 值指定当路径组的第一个路径成为活跃路径时应执行自动恢复。这可使节点在另一个节点请求故障修复时不会自动恢复。</p> <p>大于 0 的数字值指定推迟出错切换，以秒表示。</p> <p>默认值为 manual。</p>
rr_min_io	<p>指定切换到当前路径组的下一个路径前路由到该路径的 I/O 请求数。这个设置值用于运行内核为 2.6.31 之前的系统。使用新版本的系统应使用 rr_min_io_rq。默认值为 1000。</p>
rr_min_io_rq	<p>使用 request-based device-mapper-multipath 指定切换到当前路径组的下一个路径前路由到该路径的 I/O 请求数。这个设置值用于运行当前内核的系统。在使用内核 2.6.31 版本之前的系统应使用 rr_min_io。默认值为 1。</p>
rr_weight	<p>如果将其设为 priorities，就不会在调用 selector 选择下一个路径前向路径发送 rr_min_io 请求，而是由 rr_min_io 乘以路径优先权决定发送的请求数，即由 prio 功能决定。如果将其设定为 uniform，则所有路径都有相同的加权。默认值为 uniform。</p>

属性	描述
no_path_retry	<p>此属性的数字值指定了系统在禁用队列前，应该尝试使用失败路径的次数。</p> <p>值为 fail 意味着立即失败，无需排队。</p> <p>值为 queue 意味着路径固定前不会停止排队。</p> <p>默认值为 0。</p>
user_friendly_names	<p>如果将其设为 yes，即指定该系统应该使用文件 /etc/multipath/bindings 为该多路径分配一个持久且唯一的别名，格式为 mpathn。如果设定为 no，即指定该系统应使用 WWID 作为该多路径的别名。在这两种情况下，您在这里指定的数值将被您在配置文件 multipaths 部分指定的具体设备别名覆盖。默认值为 no。</p>
queue_without_daemon	<p>如果将其设为 no，则 multipathd 守护程序将会在其关闭时禁用所有设备队列。默认值为 no。</p>
flush_on_last_del	<p>如果将其设为 yes，那么 multipathd 守护程序将会在设备的最后一条路径被删除时禁用队列。默认值为 no。</p>
max_fds	<p>设定 multipath 可以打开的文件提示符以及 multipathd 守护进程的最大值。这与 ulimit -n 命令效果一致。从 Red Hat Enterprise Linux 6.3 开始，默认值为 max，该值将该系统限制到 /proc/sys/fs/nr_open。对其较早的版本，如果没有设定这个值，则使用调用进程作为打开文件提示符的最大值，通常为 1024。安全起见，如果该数值大于 1024，应将其设定为路径最大值+32。</p>
checker_timeout	<p>路径检查器和排序器执行带显式超时的 SCSI 命令的超时时间，以秒为单位。默认值为 sys/block/sdx/device/timeout 中指定的值。</p>
fast_io_fail_tmo	<p>在 FC 远程端口发现问题后，无法在那个远程端口设备中执行 I/O 前 SCSI 层要等待的时间。默认值应小于 dev_loss_tmo 值。将其设定为 off 则会禁用超时。默认值由该操作系统决定。</p>
dev_loss_tmo	<p>在 FC 远程端口发现问题后，到从该系统中删除它之前 SCSI 层要等待的时间。将其设定为无限，则会将其设定为 2147483647 秒，或者 68 年。默认值由该操作系统决定。</p>

属性	描述
hw_string_match	<p>multipath.conf 文件 devices 部分中的每个设备配置都将会创建自己的设备配置，或是修改某个内置设备配置。如果 hw_string_match 被设为 yes，那么如果某用户设备配置中的供应商、产品和修订字符串与内置设备配置中的字符串完全匹配，则该用户配置选项就会修改内置配置。否则该用户设备配置会被视为新配置。如果 hw_string_match 被设为 no，那么就会使用正则表达式匹配，而不使用字符串匹配。</p> <p>默认情况下会将 hw_string_match 设定为 no。</p>
retain_attached_hw_handler	<p>如果此参数被设为 yes，并且 SCSI 层已经为路径设备附加了硬件处理程序，那么 multipath 将不会强制设备使用 multipath.conf 文件指定的 hardware_handler。如果 SCSI 层未附加硬件处理程序，multipath 将会继续使用其配置的硬件处理程序。默认值为 no。</p>
detect_prio	<p>如果将此参数设定为 yes，multipath 将会首先检查该设备是否支持 ALUA。如果支持，则自动为该设备分配 alua 排序器；如果不支持，则按惯例确定排序器。默认值为 no。</p>
uid_attribute	<p>提供唯一路径标识符。默认值为 ID_SERIAL。</p>
force_sync	<p>（从 Red Hat Enterprise Linux 7.1 开始）如果将其设定为“yes”，则会阻止路径检查程序在异步（async）模式下运行。这意味着每次只能运行一个检查程序。这对同时运行很多 multipathd 检查程序而造成明显 CPU 负担过重时会有帮助。默认值为 no。</p>
delay_watch_checks	<p>（从 Red Hat Enterprise Linux Release 7.2 开始）如果将其设定为大于 0 的值，multipathd 守护进程将监视最近有效的路径，并执行指定数量的检查。如果在监视期间这些路径再次变为无法使用，则不会在这些路径下一次可用时就使用它们，直到连续检查使用 delay_wait_checks 指定的次数后它们都可用为止。这样可防止将那些可能不太可靠的路径在上线后立即投入使用。默认值为 no。</p>
delay_wait_checks	<p>（从 Red Hat Enterprise Linux 7.2 开始）如果将其设定为大于 0 的值，则最近重新上线的设备在由 delay_watch_checks 指定的检查次数内再次无法使用后，那么它下一次上线后就不会被标记并延迟，并在经过使用 delay_watch_checks 指定的检查次数后方可使用。默认值为 no。</p>
ignore_new_boot_devs	<p>（从 Red Hat Enterprise Linux 7.2 开始）如果设定为 yes，在引导初期该节点仍处于 initramfs 文件系统中时，multipath 不会创建任何 WWID 属于 /etc/multipath/wwids 的 initramfs copy 中的设备。当 multipath 另外尝试在手册出现时未使用 udev 规则声明的设备中进行设置时，这个功能可用于在安装过程中引导。可将这个参数设定为 yes 或者 no。如果未设置，则默认将其设定为 no。</p>

属性	描述
retrigger_tries, retrigger_delay	<p>(从 Red Hat Enterprise Linux 7.2 开始) 如果 udev 无法完成原始命令让多路径无法使用该设备, 则联合使用 retrigger_tries 和 retrigger_delay 参数, 以便 multipathd 命令可以重新激发 uevent。retrigger_tries 参数为没有完全设置的设备设定多路径尝试重新触发 uevent 的次数。retrigger_delay 参数设定两次重试之间的秒数。这两个选项均接受大于或等于 0 的数字。将 retrigger_delay 设定为 0 就是禁用重试。将 retrigger_delay 参数设定为 0 可导致在路径检查器的下一次检查中重新启动 uevent。如果没有设定 retrigger_tries 参数, 则会使用默认值 3。如果没有设定 retrigger_delay 参数, 则会使用默认值 10。</p>
new_bindings_in_boot	<p>(从 Red Hat Enterprise Linux Release 7.2 开始) 使用 new_bindings_in_boot 参数在 initramfs 文件系统中保持已被常规文件系统中绑定文件耗尽的 user_friendly_name。这样会造成问题, 因为只有重建 initramfs 文件系统时才会将 initramfs 文件系统中的 user_friendly_names 绑定与常规文件系统中的绑定同步。当将此参数设定为 no 后, 多路径不会在 initramfs 文件系统中创建任何新绑定。如果某个设备中原来没有在 /etc/multipath/bindings 的 initramfs 副本中有任何绑定, 多路径会使用其 WWID 作为别名, 而不是为其分配 user_friendly_name。之后在引导后, 该节点会挂载至常规文件系统, 多路径会为该设备分配 user_friendly_name。可将该参数设定为 yes 或者 no。如果未设定, 则默认使用 no。</p>
config_dir	<p>(从 Red Hat Enterprise Linux Release 7.2 开始) 如果设定为 "" 以外的内容, 多路径会按字母顺序搜索这些路径, 查找以 ".conf" 结尾的文件, 并从中读取配置信息, 就如同该信息位于 /etc/multipath.conf 文件中一样。这样您就会在具体机器的配置文件以外有一个主配置文件。config_dir 参数必须为 "" 或者完全限定目录名。只能在主 /etc/multipath.conf 文件中设定这个参数, 不能在由 config_dir 文件自己指定的某个文件中设定这个参数。默认值为 /etc/multipath/conf.d。</p>
deferred_remove	<p>如果设定为 yes, 则在删除最后一个路径设备时, multipathd 将会执行延期删除, 而不是常规删除。这样就会保证如果执行常规删除且操作失败时某个多路径设备正在使用中, 该设备会在最后一个用户关闭该设备时自动被删除。</p>
log_checker_err	<p>如果设定为 once, multipathd 会采用详细等级 2 记录第一个路径检查器错误。之后的所有错误都要在该设备恢复后采用详细等级 3 记录。如果设定为 always, multipathd 会一直使用详细等级 2 记录路径检查器错误。默认值为 always。</p>
skip_kpartx	<p>如果设定为 yes, kpartx 不会在该设备中自动创建分区。这样即使该设备有分区表, 也可以允许用户在不创建分区的情况下创建多路径设备。这个选项的默认值为 no。</p>

4.4. 多路径设备配置属性

表 4.2 “多路径属性”显示在 `multipath.conf` 配置文件 `multipaths` 部分中可为每个特定多路径设备设置的属性。这些属性只适用于一个指定的 `multipath`。这些默认属性可供 DM Multipath 使用，并且能覆盖 `multipath.conf` 文件中 `defaults` 和 `devices` 部分设置的属性。

表 4.2. 多路径属性

属性	描述
wwid	指定 <code>multipath</code> 属性采用的多路径设备的 WWID。这个参数在 <code>multipath.conf</code> 文件的这个部分是必须的。
alias	指定使用 <code>multipath</code> 属性的多路径设备的符号名称。如果您使用的是 <code>user_friendly_names</code> ，请必要将其设定为 <code>mpathn</code> ，这样可能会与自动分配的用户友好名称冲突，进而给出不正确的设备节点名称。
path_grouping_policy	<p>指定用于未指定路径的默认路径分组策略，可能的值包括：</p> <p>failover = 每个优先组群有一个路径</p> <p>multibus = 所有有效路径在一个优先组群中</p> <p>group_by_serial = 每个检测到的系列号有一个优先组群</p> <p>group_by_prio = 每个路径优先值有一个优先组群</p> <p>group_by_node_name = 每个目标节点名有一个优先组群</p>
path_selector	<p>指定用来决定下一个 I/O 操作所使用路径的默认算法。可能的值包括：</p> <p>round-robin 0 : 在路径组中循环每个路径，向每个路径发送同样数量的 I/O。</p> <p>queue-length 0 : 将下一组 I/O 发送到具有最少未处理 I/O 请求的路径。</p> <p>service-time 0 : 将下一组 I/O 发送到具有最短预计服务时间的路径，这是由未处理 I/O 的总量除以每个路径的相对流量决定的。</p>

属性	描述
failback	<p>管理路径组群出错切换。</p> <p>immediate 值指定立即恢复到包含活跃路径的最高级别路径组群。</p> <p>manual 值指定不需要立即恢复，只有在操作者干预的情况下会发生恢复。</p> <p>followover 值指定当路径组的第一个路径成为活跃路径时应执行自动恢复。这可使节点在另一个节点请求故障修复时不会自动恢复。</p> <p>大于 0 的数字值指定推迟出错切换，以秒表示。</p>
prio	<p>指定要获得路径优先值所需调用的默认程序及参数。例如：SPC-3 中的 ALUA 字节提供了一个可改变的 prio 值。可能的值包括：</p> <p>const：为所有路径设定优先权 1。</p> <p>emc：为 EMC 阵列生成路径优先权。</p> <p>alua：根据 SCSI-3 ALUA 设置生成路径有限级。从 Red Hat Enterprise Linux 7.3 开始，如果在设备配置中指定 prio "alua exclusive_pref_bit"，多路径将生成包含只使用 pref 字节集的路径组，并为该组设定最高优先级。</p> <p>ontap：为 NetApp 阵列生成路径优先权。</p> <p>rdac：为 LSI/Engenio RDAC 控制程序生成路径优先权。</p> <p>hp_sw：为 Compaq/HP 控制程序在激活/等待模式中生成路径优先权。</p> <p>hds：为 Hitachi HDS Modular 存储阵列生成路径优先权。</p>

属性	描述
features	<p>多路径设备的默认额外功能，使用以下格式：<code>"number_of_features_plus_arguments feature1 ..."</code>。</p> <p>features 可能的值包括：</p> <p>queue_if_no_path，与将 no_path_retry 设置为 queue 相同。欲了解使用此功能可能出现的问题，请参阅〈第 5.7 节“有 queue_if_no_path 功能的问题”〉。</p> <p>retain_attached_hw_handler：若此参数值被设为 yes，并且 SCSI 层已经为路径设备添加了硬件控制程序，multipath 将不会强制设备使用 multipath.conf 文件指定的 hardware_handler。若 SCSI 层没有添加硬件控制程序，multipath 仍然会使用已配置的硬件控制程序。默认值为 no。</p> <p>pg_init_retries <i>n</i>：路径组初始化重试，失败前最多重试 <i>n</i> 次，$1 \leq n \leq 50$。</p> <p>pg_init_delay_msecs <i>n</i>：在路径组初始化重试的间隔，等待 <i>n</i> 毫秒，$0 \leq n \leq 60000$。</p>
no_path_retry	<p>此属性的数字值指定了系统在禁用队列前，应该尝试使用失败路径的次数。</p> <p>值为 fail 意味着立即失败，无需排队。</p> <p>值为 queue 意味着路径固定前不会停止排队。</p>
rr_min_io	<p>指定切换到当前路径组的下一个路径前路由到该路径的 I/O 请求数。这个设置值用于运行内核为 2.6.31 之前的系统。使用新版本的系统应使用 rr_min_io_rq。默认值为 1000。</p>
rr_min_io_rq	<p>使用 request-based device-mapper-multipath 指定切换到当前路径组的下一个路径前路由到该路径的 I/O 请求数。这个设置值用于运行当前内核的系统。在使用内核 2.6.31 版本之前的系统应使用 rr_min_io。默认值为 1。</p>
rr_weight	<p>如果将其设为 priorities，就不会在调用 selector 选择下一个路径前向路径发送 rr_min_io 请求，而是由 rr_min_io 乘以路径优先权决定发送的请求数，即由 prio 功能决定。如果将其设定为 uniform，则所有路径都有相同的加权。</p>
flush_on_last_del	<p>如果将其设为 yes，那么当设备的最后一条路径被删除时，multipath 将会禁用队列。</p>

属性	描述
user_friendly_names	如果将其设为 yes ，即指定该系统应该使用文件 /etc/multipath/bindings 为该多路径分配一个持久且唯一的别名，格式为 mpathn 。如果设定为 no ，即指定该系统应使用 WWID 作为该多路径的别名。在这两种情况下，您在这里指定的数值将被您在配置文件 multipaths 部分指定的具体设备别名覆盖。
delay_watch_checks	（从 Red Hat Enterprise Linux Release 7.2 开始）如果将其设定为大于 0 的值， multipathd 守护进程将监视最近有效的路径，并执行指定数量的检查。如果在监视期间这些路径再次变为无法使用，则不会在这些路径下一次可用时就使用它们，直到连续检查使用 delay_wait_checks 指定的次数后它们都可用为止。这样可防止将那些可能不太可靠的路径在上线后立即投入使用。
delay_wait_checks	（从 Red Hat Enterprise Linux 7.2 开始）如果将其设定为大于 0 的值，则最近重新上线的设备在由 delay_watch_checks 指定的检查次数内再次无法使用后，那么它下一次上线后就不会被标记并延迟，并在经过使用 delay_watch_checks 指定的检查次数后方可使用。
deferred_remove	如果设定为 yes ，则在删除最后一个路径设备时， multipathd 将会执行延期删除，而不是常规删除。这样就会保证如果执行常规删除且操作失败时某个多路径设备正在使用中，该设备会在最后一个用户关闭该设备时自动被删除。
skip_kpartx	如果设定为 yes ， kpartx 不会在该设备中自动创建分区。这样即使该设备有分区表，也可以允许用户在不创建分区的情况下创建多路径设备。

以下示例显示在配置文件中为两个特定多路径设备指定的多路径属性。第一个设备的 WWID 为 **3600508b4000156d70001200000b0000**，符号名称为 **yellow**。

示例中第二个多路径设备的 WWID 为 **1DEC_____321816758474**，符号名称为 **red**。在这个示例中，**rr_weight** 属性被设为 **priorities**。

```

multipaths {
    multipath {
        wwid                3600508b4000156d70001200000b0000
        alias                yellow
        path_grouping_policy multibus
        path_selector        "round-robin 0"
        failback             manual
        rr_weight             priorities
        no_path_retry        5
    }
    multipath {
        wwid                1DEC_____321816758474
        alias                red
        rr_weight             priorities
    }
}

```

4.5. 配置文件设备

表 4.3 “设备属性”显示您可以为 `multipath.conf` 配置文件 `devices` 部分的每个独立储存设备能设置的属性。DM Multipath 会使用这些属性，除非它们被 `multipath.conf` 文件 `multipaths` 部分为包含该设备的路径所指定的属性覆盖。这些属性会覆盖 `multipath.conf` 文件 `defaults` 部分设定的属性。

多路径配置中默认包含许多支持 multipath 的设备。如需了解默认配置值（包括支持的设备）的相关信息，请运行以下命令之一。

```
# multipathd show config
# multipath -t
```

您可能不需要为这些设备修改默认值，但如果您想要修改，可以通过在配置文件中添加条目来覆盖默认值。您可以为设备复制 `multipathd show config` 命令显示的设备配置默认值，并覆盖您想要修改的值。

如需在配置文件的这部分添加没有默认自动配置的设备，您需要设置 `vendor` 和 `product` 参数。您能在 `/sys/block/device_name/device/vendor` 和 `/sys/block/device_name/device/model` 中找到这些值，其中 `device_name` 是要进行多路径操作的设备，示例如下：

```
# cat /sys/block/sda/device/vendor
WINSYS
# cat /sys/block/sda/device/model
SF2372
```

要指定的附加参数要视具体设备而定。如果设备是主动/主动模式，您通常不需要设置附加参数。您可能想要将 `path_grouping_policy` 设为 `multibus`。其它您可能需要设定的参数有 `no_path_retry` 和 `rr_min_io`，详见表 4.3 “设备属性”。

如果设备是主动/被动模式，但它会自动将路径切换到被动路径，那么您需要将检查器功能改为不需向路径发送 I/O 来测试其是否工作（否则，您的设备将一直进行出错冗余）的功能。这几乎意味着您将 `path_checker` 设为 `tur`。这对所有支持 Test Unit Ready 命令的 SCSI 设备都适用。

如果该设备需要一个特殊的命令切换路径，那么为多路径配置此设备需要硬件处理器内核模块。当前的硬件处理器是 `emc`。如果这样还不能满足您的设备，则您可能无法为多路径配置该设备。

表 4.3. 设备属性

属性	描述
<code>vendor</code>	指定 <code>device</code> 采用的存储设备的零售商名称，例如 <code>COMPAQ</code> 。
<code>product</code>	指定 <code>device</code> 属性使用的存储设备产品名，比如 <code>HSV110 (C) COMPAQ</code> 。
<code>revision</code>	指定存储设备的产品修订识别程序。
<code>product_blacklist</code>	根据产品指定用来将设备列入黑名单的正则表达式。
<code>alias_prefix</code>	这个设备类型使用的 <code>user_friendly_names</code> 前缀，而不是默认的“mpath”。

属性	描述
hardware_handler	<p>指定将在切换路径组群或者处理 I/O 错误时用来执行硬件具体动作的模块。可能的值包括：</p> <ul style="list-style-type: none"> 1 emc : EMC 存储阵列的硬件处理程序。 1 alua : SCSI-3 ALUA 阵列的硬件处理程序。 1 hp_sw : Compaq/HP 控制器的硬件处理程序。 1 rdac : LSI/Engenio RDAC 控制器的硬件处理程序。
path_grouping_policy	<p>指定用于未指定路径的默认路径分组策略，可能的值包括：</p> <ul style="list-style-type: none"> failover = 每个优先组群有一个路径 multibus = 所有有效路径在一个优先组群中 group_by_serial = 每个检测到的系列号有一个优先组群 group_by_prio = 每个路径优先值有一个优先组群 group_by_node_name = 每个目标节点名有一个优先组群
path_selector	<p>指定用来决定下一个 I/O 操作所使用路径的默认算法。可能的值包括：</p> <ul style="list-style-type: none"> round-robin 0 : 在路径组中循环每个路径，向每个路径发送同样数量的 I/O。 queue-length 0 : 将下一组 I/O 发送到具有最少未处理 I/O 请求的路径。 service-time 0 : 将下一组 I/O 发送到具有最短预计服务时间的路径，这是由未处理 I/O 的总量除以每个路径的相对流量决定的。

属性	描述
path_checker	<p>指定用于决定路径状态的默认方法，可能的值包括：</p> <p>readsector0：读取该设备的第一扇区。</p> <p>tur：在该设备中执行 TEST UNIT READY。</p> <p>emc_clariion：查询 EMC Clariion 具体 EVPD 页面 0xCO 以便决定路径。</p> <p>hp_sw：为使用 Active/Standby 固件的 HP 存储阵列检查路径状态。</p> <p>rdac：为 LSI/Engenio RDAC 存储控制器检查路径状态。</p> <p>directio：使用直接 I/O 读取第一个扇区。</p>
features	<p>多路径设备的默认额外功能，使用以下格式：<i>"number_of_features_plus_arguments feature1 ..."</i>。</p> <p>features 可能的值包括：</p> <p>queue_if_no_path，与将 no_path_retry 设置为 queue 相同。欲了解使用此功能可能出现的问题，请参阅 〈第 5.7 节“有 queue_if_no_path 功能的问题”〉。</p> <p>retain_attached_hw_handler：如果此参数被设为 yes，并且 SCSI 层已经为路径设备附加硬件处理程序，multipath 将不会强制设备使用 multipath.conf 文件指定的 hardware_handler。如果 SCSI 层未附加硬件处理程序，multipath 将会继续使用配置的硬件处理程序。</p> <p>pg_init_retries n：路径组初始化重试，失败前最多重试 <i>n</i> 次，$1 \leq n \leq 50$。</p> <p>pg_init_delay_msecs n：在路径组初始化重试的间隔，等待 <i>n</i> 毫秒，$0 \leq n \leq 60000$。</p>

属性	描述
prio	<p>指定要获得路径优先值所需调用的默认程序及参数。例如：SPC-3 中的 ALUA 字节提供了一个可改变的 prio 值。可能的值包括：</p> <p>const：为所有路径设定优先权 1。</p> <p>emc：为 EMC 阵列生成路径优先权。</p> <p>alua：根据 SCSI-3 ALUA 设置生成路径有限级。从 Red Hat Enterprise Linux 7.3 开始，如果在设备配置中指定 prio "alua exclusive_pref_bit"，多路径将生成包含只使用 pref 字节集的路径组，并为该组设定最高优先级。</p> <p>ontap：为 NetApp 阵列生成路径优先权。</p> <p>rdac：为 LSI/Engenio RDAC 控制程序生成路径优先权。</p> <p>hp_sw：为 Compaq/HP 控制程序在激活/等待模式中生成路径优先权。</p> <p>hds：为 Hitachi HDS Modular 存储阵列生成路径优先权。</p>
failback	<p>管理路径组群出错切换。</p> <p>immediate 值指定立即恢复到包含活跃路径的最高级别路径组群。</p> <p>manual 值指定不需要立即恢复，只有在操作者干预的情况下会发生恢复。</p> <p>followover 值指定当路径组的第一个路径成为活跃路径时应执行自动恢复。这可使节点在另一个节点请求故障修复时不会自动恢复。</p> <p>大于 0 的数字值指定推迟出错切换，以秒表示。</p>
rr_weight	<p>如果将其设为 priorities，就不会在调用 selector 选择下一个路径前向路径发送 rr_min_io 请求，而是由 rr_min_io 乘以路径优先权决定发送的请求数，即由 prio 功能决定。如果将其设定为 uniform，则所有路径都有相同的加权。</p>
no_path_retry	<p>此属性的数字值指定了系统在禁用队列前，应该尝试使用失败路径的次数。</p> <p>值为 fail 意味着立即失败，无需排队。</p> <p>值为 queue 意味着路径固定前不会停止排队。</p>

属性	描述
rr_min_io	指定切换到当前路径组的下一个路径前路由到该路径的 I/O 请求数。这个设置值用于运行内核为 2.6.31 之前的系统。使用新版本的系统应使用 rr_min_io_rq 。默认值为 1000。
rr_min_io_rq	使用 request-based device-mapper-multipath 指定切换到当前路径组的下一个路径前路由到该路径的 I/O 请求数。这个设置值用于运行当前内核的系统。在使用内核 2.6.31 版本之前的系统应使用 rr_min_io 。默认值为 1。
fast_io_fail_tmo	在 FC 远程端口发现问题后，无法在那个远程端口设备中执行 I/O 前 SCSI 层要等待的时间。默认值应小于 dev_loss_tmo 值。将其设定为 off 则会禁用超时。
dev_loss_tmo	在 FC 远程端口发现问题后，到从该系统中删除它之前 SCSI 层要等待的时间。将其设定为无限，则会将其设定为 2147483647 秒，或者 68 年。
flush_on_last_del	如果被设为 yes ，当设备的最后一条路径被删除时， multipathd 守护程序将会禁用队列。
user_friendly_names	如果将其设为 yes ，即指定该系统应该使用文件 /etc/multipath/bindings 为该多路径分配一个持久且唯一的别名，格式为 mpathn 。如果设定为 no ，即指定该系统应使用 WWID 作为该多路径的别名。在这两种情况下，您在这里指定的数值将被您在配置文件 multipaths 部分指定的具体设备别名覆盖。默认值为 no 。
retain_attached_hw_handler	如果此参数被设为 yes ，并且 SCSI 层已经为路径设备附加了硬件处理程序，那么 multipath 将不会强制设备使用 multipath.conf 文件指定的 hardware_handler 。如果 SCSI 层未附加硬件处理程序， multipath 将会继续使用其配置的硬件处理程序。默认值为 no 。
detect_prio	如果被设为 yes ， multipath 将会首先检查设备是否支持 ALUA。若支持，将会自动为设备分配 alua 排序器；若不支持，将会按惯例确定排序器。
uid_attribute	提供唯一路径标识符。
delay_watch_checks	（从 Red Hat Enterprise Linux Release 7.2 开始）如果将其设定为大于 0 的值， multipathd 守护进程将监视最近有效的路径，并执行指定数量的检查。如果在监视期间这些路径再次变为无法使用，则不会在这些路径下一次可用时就使用它们，直到连续检查使用 delay_wait_checks 指定的次数后它们都可用为止。这样可防止将那些可能不太可靠的路径在上线后立即投入使用。
delay_wait_checks	（从 Red Hat Enterprise Linux 7.2 开始）如果将其设定为大于 0 的值，则最近重新上线的设备在由 delay_watch_checks 指定的检查次数内再次无法使用后，那么它下一次上线后就不会被标记并延迟，并在经过使用 delay_watch_checks 指定的检查次数后方可使用。

属性	描述
deferred_remove	如果设定为 yes ，则在删除最后一个路径设备时，multipathd 将会执行延期删除，而不是常规删除。这样就会保证如果执行常规删除且操作失败时某个多路径设备正在使用中，该设备会在最后一个用户关闭该设备时自动被删除。
skip_kpartx	如果设定为 yes ，kpartx 不会在该设备中自动创建分区。这样即使该设备有分区表，也可以允许用户在不创建分区的情况下创建多路径设备。

以下示例显示了多路径配置文件的 **device** 条目。

```
# }
# device {
#   vendor    "COMPAQ  "
#   product   "MSA1000      "
#   path_grouping_policy multibus
#   path_checker tur
#   rr_weight priorities
# }
#}
```


第 5 章 DM MULTIPATH 管理及故障排除

本章提供了在运行的系统中管理 DM Multipath 的相关信息。

5.1. 使用多路径帮助程序 (MULTIPATH HELPER) 自动生成配置文件

可在 Red Hat Enterprise Linux 中使用 Multipath Helper 应用程序为多路径设备生成基本配置。该应用程序可使用自定义别名、设备黑名单和各个多路径设备的特别设定创建多路径配置。完成后，该应用程序可生成安装脚本，该脚本包含所选配置参数，并提供 `multipath.conf` 配置文件供检查。

可以在 <https://access.redhat.com/labsinfo/multipathhelper> 找到 Multipath Helper 应用程序。

5.2. 重新定义在线多路径设备大小

如果要重新定义在线多路径设备，请按以下步骤操作。

1. 重新定义物理设备大小。
2. 使用以下命令查找 LUN 路径：

```
# multipath -l
```

3. 重新定义路径大小。对于 SCSI 设备，在 `rescan` 文件中写入 1，以便让 SCSI 驱动器重新扫描，如以下命令：

```
# echo 1 > /sys/block/device_name/device/rescan
```

4. 如需重新定义多路径设备的大小，请执行 `multipathd resize` 命令：

```
# multipathd -k'resize map mpatha'
```

5. 重新定义文件系统大小（假设没有使用 LVM 或者 DOS 分区）：

```
# resize2fs /dev/mapper/mpatha
```

5.3. 将 ROOT 文件系统从单路径设备移动到多路径设备中

如果在单路径设备中安装系统，并稍后在 root 文件系统中添加了另一个路径，则需要将 root 文件系统移动到多路径设备中。本小节记录了从单路径移动到多路径设备的过程。

安装 `device-mapper-multipath` 软件包后执行以下步骤：

1. 执行以下命令创建 `/etc/multipath.conf` 配置文件，载入多路径模块并将 `multipathd` 的 `chkconfig` 设定为 `on`：

```
# mpathconf --enable
```

有关 `mpathconf` 命令输出的详情请参考第 3.1 节“设置 DM Multipath”。

2. 如果没有将 `find_multipaths` 参数配置为 `yes`，请编辑 `/etc/multipath.conf` 文件的 `blacklist` 和 `blacklist_exceptions` 部分，如第 4.2 节“配置文件黑名单”所述。

3. 要让 `multipath` 在 `root` 设备顶层发现多路径后尽快创建多路径设备，请输入以下命令。该命令还会确保 `find_multipaths` 会允许该设备，即使该设备只有一个路径。

```
# multipath -a root_devname
```

例如：如果 `root` 设备是 `/dev/sdb`，则请输入以下命令。

```
# multipath -a /dev/sdb
wwid '3600d02300069c9ce09d41c4ac9c53200' added
```

4. 为确定已正确设置配置文件，请输入 `multipath` 命令，并在输出结果中搜索使用以下格式的行。这表示该命令无法创建这个多路径设备。

```
date wwid: ignoring map
```

例如，如果设备的 WWID 为 `3600d02300069c9ce09d41c4ac9c53200`，则会看到以下输出行：

```
# multipath
Oct 21 09:37:19 | 3600d02300069c9ce09d41c4ac9c53200: ignoring map
```

5. 要使用 `multipath` 重建 `initramfs` 文件系统，请执行附带以下选项的 `dracut` 命令：

```
# dracut --force -H --add multipath
```

6. 关闭机器。
7. 配置 FC 开关以便可在该机器中看到其他路径。
8. 引导机器。
9. 查看 `root` 文件系统（`/`）是否在多路径设备中。

5.4. 将 SWAP 文件系统从单路径设备移动到多路径设备中

默认情况下会将 `swap` 设备设定为逻辑卷。将其配置为多路径设备不需要特殊的操作，只要在由逻辑卷组构成的物理卷中设定多路径即可。如果 `swap` 设备不是 LVM 卷，却使用设备名称挂载，就可能需要编辑 `/etc/fstab` 文件将其转换为正确的多路径设备名称。

1. 运行 `/sbin/multipath` 命令使用 `-v3` 选项确定 `swap` 设备的 WWID 号。该命令的输出应该在路径列表中显示该 `swap` 设备。

可以在该命令输出中看到有以下格式的行，它代表 `swap` 设备：

```
WWID H:B:T:L devname MAJOR:MINOR
```

例如：如果在 `sda` 或其分区之一中设置 `swap` 文件系统，则会在输出中看到类似如下的行：

```
===== paths list =====
...
1ATA      WDC WD800JD-75MSA3          WD-WMAM9F
1:0:0:0 sda 8:0
...

```

- 在 `/etc/multipath.conf` 文件中为 swap 设备配置别名。

```

multipaths {
    multipath {
        wwid WWID_of_swap_device
        alias swapdev
    }
}

```

- 编辑 `/etc/fstab` 文件，并使用附带多路径设备的 root 设备替换旧的设备路径。

例如：如果在 `/etc/fstab` 文件中有以下条目：

```

/dev/sda2 swap                                swap      defaults        0 0

```

可将该条目更改如下：

```

/dev/mapper/swapdev swap                    swap      defaults        0 0

```

5.5. 多路径守护进程

如果在进行多路径配置时遇到问题，则应该确定多路径守护进程正在运行，如 [第 3 章 设置 DM Multipath](#) 所述。必须运行 `multipathd` 守护进程方可使用多路径设备。

5.6. 大量 LUN 造成的问题

在某个节点中添加大量 LUN 时，使用多路径设备可明显延长 `udev` 设备管理器为其生成设备节点所消耗的时间。如果遇到这个问题，请在 `/etc/udev/rules.d/40-multipath.rules` 文件中删除以下行解决这个问题：

```

KERNEL!="dm-[0-9]*", ACTION=="add", PROGRAM==" /bin/bash -c '/sbin/lsmode | /bin/grep ^dm_multipath'", RUN+="/sbin/multipath -v0 %M:%m"

```

这行会在每次向该节点中添加块设备时让 `udev` 设备管理器运行 `multipath`。即使删除了这一行，`multipathd` 守护进程仍将自动生成多路径设备，同时在引导附带多路径 root 文件系统节点的过程中仍会调用 `multipath`。唯一的变化就是没有运行 `multipathd` 守护进程不再自动生成多路径设备，对大多数多路径用户来说应该不是个问题。

5.7. 有 QUEUE_IF_NO_PATH 功能的问题

如果使用 `features "1 queue_if_no_path"` 配置多路径设备，那么所有采用 I/O 操作的进程都将被挂起直到恢复一个或者多个路径为止。要避免这种情况，请在 `/etc/multipath.conf` 文件中设定 `no_path_retry N` 参数（其中 *N* 是该系统应该重试某个路径的次数）。

如果需要使用 `features "1 queue_if_no_path"` 选项，且遇到这里提到的问题，请使用 `dmsetup` 命令为特定 LUN 编辑运行时策略（就是说对该特定 LUN 来说所有路径都不可用）。例如：如果想要将多路径设备 `mpath2` 的策略从 `"queue_if_no_path"` 改为 `"fail_if_no_path"`，请执行以下命令。

```

dmsetup message mpathc 0 "fail_if_no_path"

```

请注意：必须指定 `mpathn` 别名而不是该路径。

5.8. 多路径命令输出

创建、修改或者列出多路径设备时，会显示当前设备设置状态，格式如下所示。

对于每个多路径设备：

```
action_if_any: alias (wwid_if_different_from_alias)
dm_device_name_if_known vendor,product size=size features='features'
hwhandler='hardware_handler' wp=write_permission_if_known
```

对与每个路径组群：

```
-+- policy='scheduling_policy' prio=prio_if_known
status=path_group_status_if_known
```

对于每个路径：

```
`- host:channel:id:lun devnode major:minor dm_status_if_known path_status
online_status
```

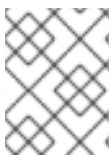
例如，多路径命令的输出可能是如下形式：

```
3600d023000000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
| `- 6:0:0:0 sdb 8:16 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
   `- 7:0:0:0 sdf 8:80 active ready running
```

如果该路径已经启用并准备好执行 I/O，那么路径的状态就是 **ready** 或者 **ghost**。如果该路径无法使用，则状态为 **faulty** 或者 **shaky**。路径的状态由 **multipathd** 守护进程根据在 **/etc/multipath.conf** 文件中定义的轮询间隔进行定期更新。

dm 状态与路径状态相似，但从内核来看，dm 有两个状态：**failed**（类似 **faulty**）和 **active**（涵盖所有其它路径状态）。少数情况下，设备的路径状态和 dm 状态会暂时不同。

online_status 的可能值为 **running** 和 **offline**。**offline** 意味着这个 SCSI 设备已被禁用。



注意

生成或者修改多路径设备时，路径组群状态、dm 设备名称、写入权限和 dm 状态是未知的。另外，功能也不一定正确。

5.9. 使用多路径命令进行多路径查询

可以使用 **multipath** 命令的 **-l** 和 **-ll** 选项来显示当前 **multipath** 配置。**-l** 选项会显示从 **sysfs** 以及设备映射器中的信息搜集到的多路径拓扑。**-ll** 选项会显示 **-l** 选项显示的信息以及系统的其他可用组件。

显示多路径配置时，可以使用 **multipath** 命令的 **-v** 选项指定三种详细等级。指定为 **-v0** 时没有输出。指定为 **-v1** 则只输出生成或者更新的路径名称。指定 **-v2** 将输出所有检测到的路径、多路径和设备映射。

以下示例显示了 `multipath -l` 命令的输出。

```
# multipath -l
3600d023000000000000e13955cc3757800 dm-1 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
| `-- 6:0:0:0 sdb 8:16 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
   `-- 7:0:0:0 sdf 8:80 active ready running
```

以下示例显示了 `multipath -ll` 命令的输出。

```
# multipath -ll
3600d023000000000000e13955cc3757801 dm-10 WINSYS,SF2372
size=269G features='0' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=enabled
| `-- 19:0:0:1 sdc 8:32 active ready running
`-+- policy='round-robin 0' prio=1 status=enabled
   `-- 18:0:0:1 sdh 8:112 active ready running
3600d023000000000000e13955cc3757803 dm-2 WINSYS,SF2372
size=125G features='0' hwhandler='0' wp=rw
|-+- policy='round-robin 0' prio=1 status=active
|  19:0:0:3 sde 8:64 active ready running
|  18:0:0:3 sdj 8:144 active ready running
```

5.10. 多路径命令选项

表 5.1 “有用的 `multipath` 命令选项” 描述了一些您可能会用到的 `multipath` 命令选项。

表 5.1. 有用的 `multipath` 命令选项

选项	描述
<code>-l</code>	显示来自 <code>sysfs</code> 和设备映射器的当前多路径配置。
<code>-ll</code>	显示来自 <code>sysfs</code> 、设备映射器以及系统中其他所有可用组件的当前多路径配置。
<code>-f device</code>	删除命名的多路径设备。
<code>-F</code>	删除所有不使用的多路径设备。
<code>-w device</code>	从 <code>wwids</code> 文件中删除指定设备的 <code>wwid</code> 。
<code>-W</code>	重新设定 <code>wwids</code> 文件使其只包含当前 <code>multipath</code> 设备。

5.11. 使用 `DMSETUP` 命令确定设备映射器条目

可以使用 `dmsetup` 找出哪个设备映射器条目与多路径的设备映射。

以下命令显示所有设备映射器设备及其主、副号码。副号码确定 dm 设备的名称。例如：副号码 3 与多路径的设备 `/dev/dm-3` 对映。

```
# dmsetup ls
mpathd (253:4)
mpathep1 (253:12)
mpathfp1 (253:11)
mpathb (253:3)
mpathgp1 (253:14)
mpathhp1 (253:13)
mpatha (253:2)
mpathh (253:9)
mpathg (253:8)
VolGroup00-LogVol01 (253:1)
mpathf (253:7)
VolGroup00-LogVol00 (253:0)
mpathe (253:6)
mpathbp1 (253:10)
mpathd (253:5)
```

5.12. MULTIPATHD 命令

可使用 `multipathd` 命令管理 `multipathd` 守护进程。有关可用的 `multipathd` 命令的详情，请查看 `multipathd(8)` 手册页。

某些 `multipathd` 命令包含 `format` 选项，后接通配符。可以使用以下命令显示可用的通配符清单。

```
# multipathd show wildcards
```

从 Red Hat Enterprise Linux release 7.3 开始，`multipathd` 命令支持新格式命令，以便显示多路径状态以及使用“raw”格式版本的路径。在 raw 格式中，不会显示标头，且不会添加字段以便将标头与该列对其。反之，该字段以指定的格式显示。这样就可以更好地使用输出结果编写脚本。现在可使用 `multipathd show wildcards` 命令显示格式字符串中的通配符。

要按照 `multipathd` 命令显示 `multipathd` 监控的多路径设备，使用带多路径通配符的格式字符串，可以是常规格式，也可以是 raw 格式。

```
list|show maps|multipaths format $format
list|show maps|multipaths raw format $format
```

以下 `multipathd` 命令显示 `multipathd` 监控的多路径设备，使用带多路径通配符的格式字符串，可以是常规格式，也可以是 raw 格式。

```
list|show paths format $format
list|show paths raw format $format
```

以下命令显示 `multipathd show maps` 中非 raw 和 raw 格式之间的不同。注：在 raw 格式中没有标头，只在每个列之间有一个空格。

```
# multipathd show maps format "%n %w %d %s"
name    uuid                                sysfs vend/prod/rev
mpathc 360a98000324669436c2b45666c567942 dm-0 NETAPP,LUN
```

```
# multipathd show maps raw format "%n %w %d %s"  
mpathc 360a98000324669436c2b45666c567942 dm-0 NETAPP,LUN
```

5.13. 使用 MULTIPATHD 互动控制台进行故障排除

multipathd -k 命令是 **multipathd** 守护程序的交互式界面。输入这一命令将进入交互式多路径控制台。执行此命令后，就可以输入 **help** 获取可用命令列表。可输入交互式命令，或者按 **CTRL-D** 退出。

multipathd 交互式控制台可用来在系统出问题后进行故障排除。例如：以下命令会在退出控制台前显示多路径配置，其中包括默认配置。

```
# multipathd -k  
> > show config  
> > CTRL-D
```

以下命令确定多路径已经识别了所有对 **multipath.conf** 的修改。

```
# multipathd -k  
> > reconfigure  
> > CTRL-D
```

使用以下命令组确定路径检查器是否正常工作。

```
# multipathd -k  
> > show paths  
> > CTRL-D
```

5.14. 删除软件包后清除多路径文件

如果不小心删除了 **device-mapper-multipath rpm** 文件，请注意这不会删除 **/etc/multipath.conf**、**/etc/multipath/bindings** 和 **/etc/multipath/wwids** 文件。但可能需要在后续 **device-mapper-multipath** 软件包安装中手动删除这些文件。

附录 A. 修订历史

修订 0.4-3.2 与 XML 源 0.4-3 版本同步的翻译文件	Mon Jun 18 2018	Ludek Janda
修订 0.4-3.1 与 XML 源 0.4-3 版本同步的翻译文件	Mon Nov 28 2016	Terry Chuang
修订 0.4-3 7.3 GA 发布版本。	Mon Oct 24 2016	Steven Levine
修订 0.4-1 准备 7.3 Beta 发布文档	Mon Aug 15 2016	Steven Levine
修订 0.3-6 准备 7.2 GA 发布文档	Mon Nov 9 2015	Steven Levine
修订 0.3-3 准备 7.2 Beta 发布文档	Wed Aug 19 2015	Steven Levine
修订 0.2-7 7.1 GA 发行版本	Mon Feb 16 2015	Steven Levine
修订 0.2-6 7.1 Beta 发行版本	Thu Dec 11 2014	Steven Levine
修订 0.1-22 7.0 GA 发行版本	Mon Jun 2 2014	Steven Levine
修订 0.1-10 7.0 Beta 更新	Wed Apr 9 2014	Steven Levine
修订 0.1-3 7.0 pre-Beta 版本	Tue Nov 26 2013	Steven Levine
修订 0.1-1 衍生自 Red Hat Enterprise Linux 6 版本的文件	Wed Jan 16 2013	Steven Levine

索引

符号

/etc/multipath.conf 软件包, [设置 DM Multipath](#)

主动/主动配置

图解, [DM Multipath 概述](#)

定义, [DM Multipath 概述](#)

主动/被动配置

图示, [DM Multipath 概述](#)

定义, [DM Multipath 概述](#)

全球识别符 (WWID), [多路径设备识别符](#)

功能, 新的和更改的, [新的和更改的功能](#)

多路径 root 文件系统, [将 root 文件系统从单路径设备移动到多路径设备中](#)

多路径 swap 文件系统, [将 swap 文件系统从单路径设备移动到多路径设备中](#)

多路径守护进程 (multipathd), [多路径守护进程](#)

多路径设备, [多路径设备](#)

LVM 物理卷, [逻辑卷中的多路径设备](#)

逻辑卷, [逻辑卷中的多路径设备](#)

存储阵列

添加, [配置存储设备](#), [配置文件设备](#)

存储阵列支持, [存储阵列支持](#)

故障转移, [DM Multipath 概述](#)

本地磁盘, 忽略, [在生成多路径设备时忽略逻辑磁盘](#)

概述

功能, 新的和更改的, [新的和更改的功能](#)

设备

添加, [配置存储设备](#), [配置文件设备](#)

设备名称, [多路径设备识别符](#)

设置

DM Multipath, [设置 DM Multipath](#)

配置

DM Multipath, [设置 DM Multipath](#)

配置文件

alias 参数, [多路径设备配置属性](#)

alias_prefix 参数, [配置文件设备](#)

checker_timeout 参数, [配置文件默认设置](#)

config_dir 参数, [配置文件默认设置](#)

[deferred_remove](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

[delay_wait_checks](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

[delay_watch_checks](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

[detect_prio](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#)

[dev_loss_tmo](#) 参数, [配置文件默认设置](#), [配置文件设备](#)

[failback](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

[fast_io_fail_tmo](#) 参数, [配置文件默认设置](#), [配置文件设备](#)

[features](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

[flush_on_last_del](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

[force_sync](#) 参数, [配置文件默认设置](#)

[hardware_handler](#) 参数, [配置文件设备](#)

[hw_string_match](#) 参数, [配置文件默认设置](#)

[ignore_new_boot_devs](#) 参数, [配置文件默认设置](#)

[log_checker_err](#) 参数, [配置文件默认设置](#)

[max_fds](#) 参数, [配置文件默认设置](#)

[new_bindings_in_boot](#) 参数, [配置文件默认设置](#)

[no_path_retry](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

[path_checker](#) 参数, [配置文件默认设置](#), [配置文件设备](#)

[path_grouping_policy](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

[path_selector](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

[polling-interval](#) 参数, [配置文件默认设置](#)

[prio](#) 参数, [配置文件默认设置](#), [配置文件设备](#)

[product](#) 参数, [配置文件设备](#)

[product_blacklist](#) 参数, [配置文件设备](#)

[queue_without_daemon](#) 参数, [配置文件默认设置](#)

[reassign_maps](#) 参数, [配置文件默认设置](#)

[retain_attached_hw_handler](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#)

[retrigger_delay](#) 参数, [配置文件默认设置](#)

[retrigger_tries](#) 参数, [配置文件默认设置](#)

[revision](#) 参数, [配置文件设备](#)

[rr_min_io](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#)

[rr_weight](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

[skip_kpartx](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

[uid_attribute](#) 参数, [配置文件默认设置](#), [配置文件设备](#)

[user_friendly_names](#) 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

[vendor](#) 参数, [配置文件默认设置](#), [配置文件设备](#)

[wwid](#) 参数, [多路径设备配置属性](#)

[概述](#), [配置文件概述](#)

[黑名单](#), [配置文件黑名单](#)

重新定义多路径设备大小, [重新定义在线多路径设备大小](#)

黑名单

WWID, [根据 WWID 将设备列入黑名单](#)
设备名称, [根据设备名称将设备列入黑名单](#)
设备类型, [根据设备类型将其加入黑名单](#)
配置文件, [配置文件黑名单](#)
默认设备, [根据设备名称将设备列入黑名单](#)

A

alias 参数, [多路径设备配置属性](#)
配置文件, [多路径设备识别符](#)

alias_prefix 参数, [配置文件设备](#)

B

blacklist_exceptions 部分
multipath.conf 文件, [黑名单之外的设备](#)

C

checker_timeout 参数, [配置文件默认设置](#)
config_dir 参数, [配置文件默认设置](#)

D

defaults 部分
multipath.conf 文件, [配置文件默认设置](#)

deferred_remove 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)
delay_wait_checks 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)
delay_watch_checks 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)
detect_prio 参数, [配置文件默认设置](#), [多路径设备配置属性](#)
dev/mapper 目录, [多路径设备识别符](#)

device-mapper-multipath 软件包, [设置 DM Multipath](#)

devices 部分
multipath.conf 文件, [配置文件设备](#)

dev_loss_tmo 参数, [配置文件默认设置](#), [配置文件设备](#)

DM Multipath

冗余, [DM Multipath 概述](#)
和 LVM, [逻辑卷中的多路径设备](#)
定义, [设备映射器多路径](#)
故障转移, [DM Multipath 概述](#)
概述, [DM Multipath 概述](#)
组件, [DM Multipath 组件](#)
设备, [多路径设备](#)

设备名称, [多路径设备识别符](#)

设置, [设置 DM Multipath](#)

设置, [概述, DM Multipath 设置概述](#)

配置, [设置 DM Multipath](#)

配置文件, [DM Multipath 配置文件](#)

dm-multipath kernel 模块, [DM Multipath 组件](#)

dm-n 设备, [多路径设备识别符](#)

dmsetup 命令, [确定设备映射器条目, 使用 dmsetup 命令确定设备映射器条目](#)

F

failback 参数, [配置文件默认设置, 多路径设备配置属性, 配置文件设备](#)

fast_io_fail_tmo 参数, [配置文件默认设置, 配置文件设备](#)

features 参数, [配置文件默认设置, 多路径设备配置属性, 配置文件设备](#)

flush_on_last_del 参数, [配置文件默认设置, 多路径设备配置属性, 配置文件设备](#)

force_sync 参数, [配置文件默认设置](#)

H

hardware_handler 参数, [配置文件设备](#)

hw_string_match 参数, [配置文件默认设置](#)

I

ignore_new_boot_devs 参数, [配置文件默认设置](#)

initramfs

启动 multipath, [在 initramfs 文件系统中设置启动 Multipath](#)

K

kpartx 命令, [DM Multipath 组件](#)

L

log_checker_err 参数, [配置文件默认设置](#)

LVM 物理卷

多路径设备, [逻辑卷中的多路径设备](#)

lvm.conf 文件, [逻辑卷中的多路径设备](#)

M

max_fds 参数, [配置文件默认设置](#)

multipathconf 命令, [DM Multipath 组件](#)

Multipath Helper, [使用多路径帮助程序 \(Multipath Helper\) 自动生成配置文件](#)

multipath 命令, [DM Multipath 组件](#)

查询, [使用多路径命令进行多路径查询](#)

输出结果, [多路径命令输出](#)

选项, [多路径命令选项](#)

`multipath.conf` 文件, [存储阵列支持](#), [DM Multipath 配置文件](#)

`blacklist_exceptions` 部分, [黑名单之外的设备](#)

`defaults` 部分, [配置文件默认设置](#)

`devices` 部分, [配置文件设备](#)

`multipaths` 部分, [多路径设备配置属性](#)

`multipathd`

互动控制台, [使用 `multipathd` 互动控制台进行故障排除](#)

命令, [使用 `multipathd` 互动控制台进行故障排除](#)

`multipathd start` 命令, [设置 DM Multipath](#)

`multipathd` 守护进程, [DM Multipath 组件](#)

`multipaths` 部分

`multipath.conf` 文件, [多路径设备配置属性](#)

N

`new_bindings_in_boot` 参数, [配置文件默认设置](#)

`no_path_retry` 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

P

`path_checker` 参数, [配置文件默认设置](#), [配置文件设备](#)

`path_grouping_policy` 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

`path_selector` 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

`polling_interval` 参数, [配置文件默认设置](#)

`prio` 参数, [配置文件默认设置](#), [配置文件设备](#)

`product` 参数, [配置文件设备](#)

`product_blacklist` 参数, [配置文件设备](#)

Q

`queue_without_daemon` 参数, [配置文件默认设置](#)

R

`reassign_maps` 参数, [配置文件默认设置](#)

`retain_attached_hw_handler` 参数, [配置文件默认设置](#), [多路径设备配置属性](#)

`retrigger_delay` 参数, [配置文件默认设置](#)

`retrigger_tries` 参数, [配置文件默认设置](#)

`revision` 参数, [配置文件设备](#)

`root` 文件系统, [将 `root` 文件系统从单路径设备移动到多路径设备中](#)

`rr_min_io` 参数, [配置文件默认设置](#), [多路径设备配置属性](#)

`rr_weight` 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

S

skip_kpartxr 参数, [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)
swap 文件系统, [将 swap 文件系统从单路径设备移动到多路径设备中](#)

U

uid_attribute 参数, [配置文件默认设置](#), [配置文件设备](#)

user_friendly_names 参数, [多路径设备识别符](#), [配置文件默认设置](#), [多路径设备配置属性](#), [配置文件设备](#)

V

vendor 参数, [配置文件设备](#)

verbosity 参数, [配置文件默认设置](#)

W

wwid 参数, [多路径设备配置属性](#)